Carlos Morato
Department of Mechanical Engineering &
Institute for Systems Research,
University of Maryland,
College Park, MD 20742
e-mail: cmorato@umd.edu

Krishnanand N. Kaipa
Department of Mechanical Engineering &
Institute for Systems Research,
University of Maryland,
College Park, MD 20742
e-mail: kkrishna@umd.edu

Boxuan Zhao
Department of Mechanical Engineering &
Institute for Systems Research,
University of Maryland,
College Park, MD 20742
e-mail: zhaoboxuan@gmail.com

Satyandra K. Gupta[1]
Department of Mechanical Engineering &
Institute for Systems Research,
University of Maryland,
College Park, MD 20742
e-mail: skgupta@umd.edu

Toward Safe Human Robot Collaboration by Using Multiple Kinects Based Real-Time Human Tracking

We present a multiple Kinects based exteroceptive sensing framework to achieve safe human-robot collaboration during assembly tasks. Our approach is mainly based on a real-time replication of the human and robot movements inside a physics-based simulation of the work cell. This enables the evaluation of the human-robot separation in a 3D Euclidean space, which can be used to generate safe motion goals for the robot. For this purpose, we develop an N-Kinect system to build an explicit model of the human and a roll-out strategy, in which we forward-simulate the robot's trajectory into the near future. Now, we use a precollision strategy that allows a human to operate in close proximity with the robot, while pausing the robot's motion whenever an imminent collision between the human model and any part of the robot is detected. Whereas most previous range based methods analyzed the physical separation based on depth data pertaining to 2D projections of robot and human, our approach evaluates the separation in a 3D space based on an explicit human model and a forward physical simulation of the robot. Real-time behavior ($\approx 30\,Hz$) observed during experiments with a 5 DOF articulated robot and a human safely collaborating to perform an assembly task validate our approach.
[DOI: 10.1115/1.4025810]

1 Introduction

Robots excel at performing tasks—welding, component soldering, bolting, packaging—requiring speed, repeatability, and high payload capabilities. However, humans are better at manipulation of a wide range of parts without using special fixtures; they also have a natural ability to handle unexpected situations on the shop floor. Therefore, collaborative frameworks in which humans and robots share the workspace and closely work together to perform manufacturing tasks can lead to increased levels of productivity.

Safety is one of the primary challenges encountered while trying to introduce robots into anthropic environments [1–3]. Traditionally, safety in work cells is ensured by caging a robot with either a physical [4] or virtual [5] barrier and sequencing the roles of the robot and the human; that is, the robot is rendered inoperative whenever a human enters the robot's work cell to perform his/her task. However, this segregation paradigm leaves no scope to realize the proposed benefits of human-robot collaboration (HRC).

Strategies to achieve safe HRC can be broadly divided into two categories: Precollision [6–12] and postcollision [1,13,14]. The former problem deals with devising controllers that allow the robot to prevent imminent collisions with a human. However, the latter aims to reduce the impact/injury after an unexpected human-robot collision has occurred. One example is a human-friendly robot designed by Shin et al. [14]. In this paper, we limit the scope of our literature review to precollision strategies as the methods presented in this paper belong to this category.

The underlying principle of most precollision methods consists of calculating the physical separation between the robot and the human, tracking the changes in the separation, and enabling the robot to take preventive actions whenever the separation is below a specified threshold. Separation monitoring in shared workspaces has been identified as one of the important aspects for which performance metrics are appearing in the recent literature [15]. Successful deployment of human robot collaboration systems involves a proper integration between low-loop control loops and high-level planners [16]. In this context, separation monitoring also provides the perceptual feedback required to implement expressive temporal planners [17], which integrate sharable resource management into plan generation. This feedback can also be used in conjunction with assembly planners [18,19] and instruction visualization tools [20–22] to modify assembly plans and assembly instructions appropriately.

Note that the separation of interest is not a simple Euclidean distance between two points, since a collision can occur between any part of the human and any part of the robot during a collaborative task. Moreover, certain parts of the human (robot) have more probability of collision with the robot (human) than certain others. For example, consider a human and a desktop-robot manipulator working in close proximity with respect to each other. In this HRC scenario, the human's hands are exposed to the arms of the robot with a higher frequency than that of his/her trunk; the human's legs never come into contact with the robot as they always operate below the surface of the assembly table.

All these issues raise important questions of how to model the robot-human separation, how to design sensing methods to accurately measure the model variables, and how to incorporate the resulting data into robot control for ensuring safety in the work cell. Previous approaches that address these challenges mainly differ from one another depending on (a) how the human's motion is accounted for; that is, whether the human is tracked by using an explicit 3D human model or he/she is treated as equivalent to other obstacles in the work cell, (b) if a human model is used, then what sensing method is used to build the model, and (c) what control algorithm is used to prevent collisions between the human and the robot during the course of their collaborations.

Recent advances in computer game interfaces have enabled their use as tools for interaction with robots. For example, Smith and Christensen [23] presented a method to use wiimote controller to track human input based on human motion models. Similarly,

[1]Corresponding author.

Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received July 27, 2013; final manuscript received October 20, 2013; published online January 22, 2014. Assoc. Editor: Joshua D. Summers.

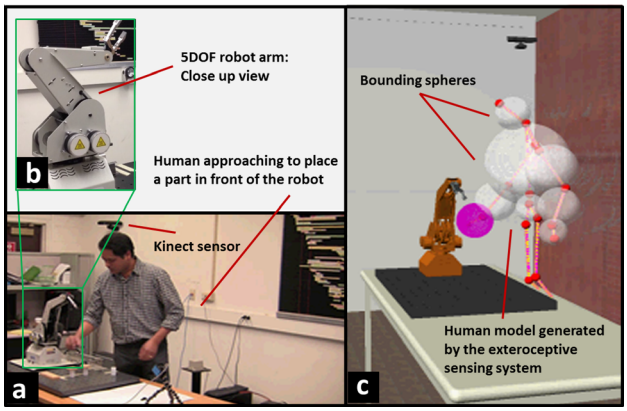

Fig. 1 Overall system overview: (*a*) Work cell used to evaluate human-robot interaction. (*b*) 5 DOF robot used for the experiments. (*c*) Physical simulation used to evaluate the interference between the human and the robot in real-time.

Kinect is another low-cost sensing device that is recently being used for HRC applications [8].

In this paper, we present a multiple Kinects based exteroceptive sensing framework to achieve safe human-robot collaboration during assembly tasks. An overview of the overall system is shown in Fig. 1. A preliminary implementation of the system was presented in Ref. [24]. Our approach consists of a real-time replication of the human and robot movements inside a physics-based simulation of the work cell. This enables the evaluation of the interactions between them in a three dimensional Euclidean space, which can be used to generate safe motion goals for the robot. First, we develop an *N*-Kinect based framework that builds an explicit model of the human in near real-time. In particular, the sensing system consists of multiple Microsoft Kinects mounted at various points on the periphery of the work cell. Usage of multiple Kinects accounts for problems caused by occlusion. Each Kinect monitors the human and outputs a 20-joint human model. Data acquired from all the Kinects are fused in a filtering scheme to obtain a refined estimate of the human's motion. Second, the generated human model is augmented by approximating pairs of neighboring joints with dynamic bounding spheres that move as a function of the movements performed by the human in real-time. Third, we implement a *roll-out* strategy in a physics-based engine, where we forward-simulate the robot's trajectory into the near future, creating a temporal set of robot's postures for the next few seconds; now, we check whether any of these postures collides with one of the bounding spheres of the human model. Fourth, we use a precollision strategy that allows a human to operate in close proximity with the robot, while pausing the robot's motion whenever an imminent collision between the human and any part of the robot is detected. Whereas most previous range based methods analyzed the physical separation based on depth data pertaining to 2D projections of robot and human, our approach is one of the first successful attempts to evaluate human-robot interference in a three dimensional Euclidean space based on an explicit human model and a forward physical simulation of the robot. Real-time behavior ($\approx$ 30 Hz) observed during experiments with a 5 DOF articulated robot and a human safely interacting to perform a shared assembly task validate the effectiveness of our approach.

2 Related Work

We have identified two families of precollision approaches that significantly differ from each other in their underlying philosophies and, consequently, in their implementation techniques. The first line of research direction treats the problem in a two dimensional Euclidean space by working with the projections of the human and the robot onto a 2D range-image plane [12,7,8,25–27]. However, the second one analyzes the problem directly in a three dimensional Euclidean space by using explicit 3D models for the human and the robot [6,28]. We briefly describe these approaches and compare them to the methods presented in this paper.

2.1 Interaction Analysis in 2D Euclidean Space. Schiavi et al. [12] presented an approach to generate safe robot motion goals based on human presence/position detection in the work cell. The intersection between the robot and the human was determined based on analysis in a 2D plane. The human was not explicitly modeled; instead, was treated as a general moving obstacle and a corresponding depth image was generated by using a stereo camera based range sensing system. The robot's 3D-occupancy[2] with respect to the global reference frame was computed from its 3D CAD model and kinematics. Next, the occupancy data were projected onto the camera image plane, giving rise to the depth image of the robot. Now, an intersection between the two projections was used as a necessary condition for a collision between the robot and the obstacle. That is, the robot and the obstacle are physically separated in 3D, if their respective projections on the image plane do not intersect. However, if the projections intersect with each other, then there is a possibility that the robot and the obstacle are in collision or may collide with each other in the near future. In a test result, a physical robot used the proposed method to safely navigate around a human hand and reach the target configuration.

However, in this work, a single depth sensor is used to monitor the environment, which leads to lack of information in the blind zones of the sensor. Moreover, when parts of the obstacle are occluded by the robot, the obstacle depth information at the corresponding pixel locations is not available, which could lead to a system failure. In order to address the problem of occlusions, Flacco and De Luca [7] extended the approach in Ref. [12] to multiple depth sensors. The collision detection performance was maximized by solving an optimal sensor placement problem that was formulated by using a probabilistic framework. In particular, they decomposed the work cell into discrete cells and derived expressions for probabilities of each cell falling in occlusion and unobserved regions as a function of pose parameters of the sensors. Now, a cost function, to be minimized for optimal sensor placement, was defined as a weighted sum of the derived probabilities. The authors used numerical simulations to compute optimal sensor placements for the cases of one, two, and three sensors. However, their work was limited to a theoretical treatment and computer simulations. No physical experiments were used to evaluate the efficacy of their approach. Later, Flacco et al. [8] presented a slightly different approach, in which the distances between the robot and the obstacles were computed directly from depth data obtained from a Kinect based range sensor, instead of projecting the depth data into a robot-oriented space. These computed distances were then used in a potential field based technique that allowed the robot to avoid collisions with humans and other moving obstacles. The authors reported results from physical experiments in which a 7 DOF KUKA Light-Weight-Robot IV safely avoided collisions with a human in the work cell.

Fischer et al. [26] managed the user occlusion in an augmented scene by using depth maps originated from time-of-flight sensors. In addition, Valentini [27] developed a tracking system using a Kinect sensor for computing both global geometry occlusion and natural interaction with objects. They both managed the entire depth map in real-time, without simplification, where the tracking of the entire geometry is useful for avoiding misinterpretation of joint location using the single Kinect sensor.

2.2 Interaction Analysis in 3D Euclidean Space. Balan and Bone [6] addressed the human-robot collision problem by using sphere-based geometric models for the human and robot. Their

[2] Collection of all points in the work cell that are occupied by the robot.

Table 1 Kinect specifications used in the sensing design

Parameter	Specifications
Output	20-joint human skeleton model; 3D position coordinates for each joint given in meters
Operating range	0.8 to 3.5 m
Horizontal field of view	57 deg
Vertical field of view	43 deg
Spatial resolution	0.003 m
Depth resolution	0.01 m
Kinect SDK Version	V1.6

algorithm selected search directions that balanced between the two objectives of robot approaching the target configuration and maximizing its distance to the human throughout its motion. The robot's motion was predicted by using a transfer function model of its time response at the joint level. The human's motion was predicted at the sphere level by using a weighted mean of past velocities. As a test scenario, the authors developed a simulation of a human walking toward a moving Puma robot arm. The authors used captured human motion data to create a realistic animation. They used Monte Carlo simulations, consisting of 1000 random human walking paths passing through the robot workspace, to validate their approach. However, no real robot experiments were conducted.

Najmaei and Kermani [11,28] also addressed the human-robot collision problem by incorporating explicit 3D modeling of the human into their approach to safe HRC. For this purpose, they developed floor mat, a sensing system comprising a grid of nodes that got activated under human body weight. The human localization was derived based on which clusters of nodes were activated as a function of time. This information, along with the average human body dimensions, was used to obtain a human model, which was then represented as a moving obstacle in the human-robot interaction framework.

2.3 Observations. In all the above approaches to safe HRC that used range or camera based systems to detect humans, the human-robot separation was analyzed in a 2D Euclidean space by using the depth information extracted from the camera images. However, our approach performs the analysis in a 3D Euclidean space, similar to Refs. [6,28], by working with an explicit 3D human model generated from Kinect and a forward 3D simulation of the robot's motion in a physics-based virtual environment. Whereas the 2D based approaches discussed above were proposed to overcome the speed limitations of 3D space analysis based techniques [12], we show that our approach, which belongs to the latter category, still achieves satisfactory real time performance. Also, we develop a multiple Kinect based framework in order to take care of occlusions as opposed to using a single sensor in Refs. [12,8].

3 Real-Time Human Motion Tracking

Tracking of the human inside the work cell is achieved by generating a skeleton-like model of the human and by estimating the 3D positions of its joints in order to determine the human's movements. For this purpose, we use an *N*-Kinect based exteroceptive sensing system, which consists of multiple Kiencts mounted at various points on the periphery of the work cell. Each Kinect monitors the human and outputs a 20-joint human model (Fig. 2) in its local reference frame. Positional data from all the Kinects are fused in a filtering scheme in order to obtain a refined human model in the global frame of reference.

Instead of processing the entire depth map, our sensing system works with a 20 DOF human model. This limited number of joints used to describe the human pose ensure the real-time operation of the framework, the scalability, and the latency free sensor fusion by reducing the number of variables to be processed and by reducing the amount of data to be transferred. Unlike previous gesture-based human tracking systems, usage of the Kinect does not require the human to wear any sensing-related devices. The specifications of the Kinect are shown in Table 1.

3.1 Exteroceptive Sensing Configuration. Design of the sensing configuration, given the work volume shared by the robot and the human, is mainly influenced by factors like shape of the workspace, number of sensors, placement of sensors, and presence of dead zones. We carry out a systematic experimental analysis of these factors in order to characterize the performance of the sensing system. In general, our objective is to achieve an optimal coverage of the workspace by maximizing the number of fully tracked joints, while minimizing the number of sensors used.

3.1.1 Workspace Analysis. Our framework considers a *N*-Kinect based sensing system, where *N* is the number of Kinects required to fully cover the work volume. The shape of the work volume considered in the experiments is cylindrical by nature. Therefore, there is no need for a Kinect to be placed directly above the robot. However, there is a need for multiple Kinects to be placed radially surrounding the periphery of the work cell. The exact placement of each Kinect in the radial direction and the angular separation between two neighboring Kiencts is guided by the operating range and the horizontal field of view of the Kinect (Table 1) and the dimensions of the work cell ($4.72\,\text{m} \times 3.2\,\text{m} \times 2.7\,\text{m}$). The height and the pitch[3] ($= -20$ deg) of each Kinect are selected such that a human with hands in a upright position is within the vertical field of view of the Kinect.

3.1.2 Number of Kinects. Intuitively, coverage increases with an increase in the number of Kinects. However, the signals from multiple Kinects tend to interfere with each other. In particular, the infrared-ray pattern generated by the Kinect is not modulated in a way that the Kinect can recognize its own pattern; thereby, one Kinect could cast a ray that another Kinect defines as its own and hence incorrectly estimates the distance. Therefore, the number of Kinects must be chosen such that the coverage is maximized, while the interference between two neighboring Kinects is minimized.

We studied these effects by conducting the following experiment: We placed one Kinect at an appropriate distance to the center of the work cell and logged the values of metrics like workspace coverage,[4] assembly cell coverage,[5] implicit rotation, and the number of fully tracked human joints. Next, we incrementally added a new Kinect at some angular separation to the previous Kinect (but at the same distance to the work cell center as that of the previous one) and recorded the readings again. A typical sensor arrangement with multiple Kinects mounted on the periphery of the work cell is shown in Fig. 3. The yaw[6] ($= 50$ deg) of each Kinect is fixed at an angle such that the Kinect axis makes a small offset with the nearest diagonal of the work cell. This reduces the overlap with the Kinect facing diametrically opposite to it, thereby, increasing the net coverage due to the two Kinects.

Table 2 shows how the values of the metrics mentioned above varied as a function of the angle between two neighboring Kinects and the number of Kinects used up to the current step. From these experiments, we find that four Kinects mounted on the corners of the work cell are sufficient to cover the workspace. Note that there is no additional benefit in using more than four Kinects for the given work cell.

[3] Angle between the sensor axis and the horizontal plane.
[4] Ratio of area covered by the Kinect and total area of the workspace.
[5] Ratio of workspace coverage and the total area of the work cell.
[6] Angle between the Kinect axis and the side wall.

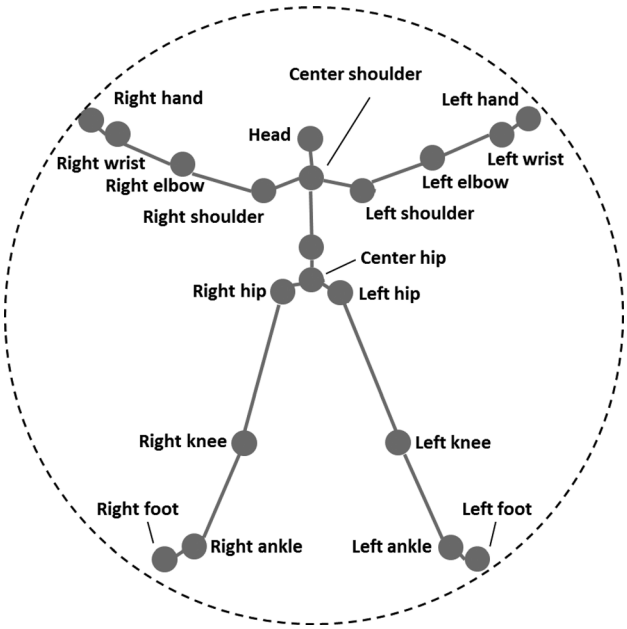

Fig. 2 The Microsoft Kinect directly outputs a 20-joint model of a human observed in a 3D scene

Table 2 Coverage as a function of number of Kinects

Number of Kinects	1	2	3	4	5	6
Assembly cell coverage ($\approx$ %)	20	35	55	85	88	90
Workspace coverage ($\approx$ %)	25	50	75	100	100	100
Implicit rotation (degrees)	90	270	360	360	360	360
Number of fully tracked joints	4	8	20	20	20	20

3.1.3 Dead zones. Dead zones correspond to regions which have either poor or no coverage. With respect to each Kinect (Fig. 3), the blue-colored region is fully covered and the red-colored region is poorly covered. Accordingly, from Fig. 3, the red- and white-colored regions are the dead zones of the work cell. These sensing failures are handled by choosing the number of Kinects and their postures such that the workspace shared by the robot and the human is a proper subset of the union of the volumes covered by all the Kinects. From Fig. 3, note that the workspace marked as the dotted rectangle completely falls within the net coverage of all the Kinects.

3.2 Human Model Estimation. Each joint position of the human model generated by a Kinect p_{ij} (where i and j are the Kinect and joint indices, respectively) is estimated by using a separate discrete Kalman filter. This results in a set of twenty local filters corresponding to twenty joints for each Kinect. Next, the resulting estimates of each joint j from all Kinects are used as inputs to a particle filter. This results in a set of twenty particle filters used to obtain improved estimates of all twenty joints.

The Kinect software cannot handle data from multiple Kinects. Therefore, individual models obtained from different Kinects are integrated via a communication architecture based on User Datagram Protocol (UDP). A client computer reads the positional data of the human model from each Kinect and transforms it into global coordinates. Next, the joint-position estimates from all 20×4 local filters are sent to the server, in which the particle filters are implemented.

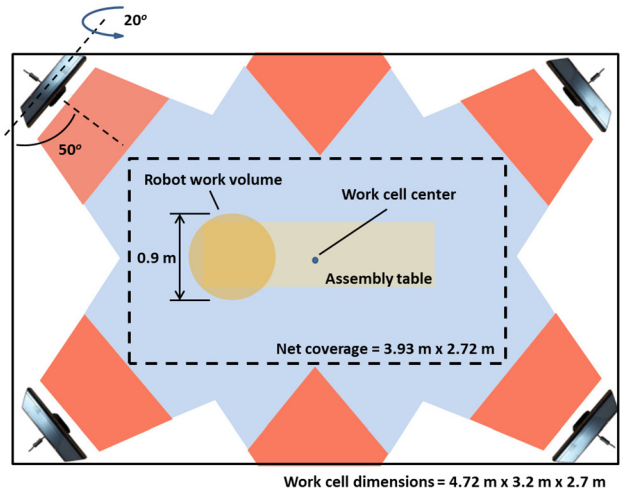

Fig. 3 Coverage (horizontal projection) obtained by using four Kinect sensors. The blue-color regions are fully covered. Red- and white-colored regions represent the dead regions of the work cell.

3.2.1 Local Filter. We derive an approximate model of human motion as follows. Let $p_j = (x_j, y_j, z_j), \dot{p}_j = (\dot{x}_j, \dot{y}_j, \dot{z}_j)$, and $\ddot{p}_j = (\ddot{x}_j, \ddot{y}_j, \ddot{z}_j)$ represent the position, velocity, and acceleration of each joint j. Writing the Taylor series expansion for position and velocity along the x-axis, we have

$$\begin{aligned} x_j(k+1) &= x_j(k) + \Delta T \dot{x}_j(k) + \frac{\Delta T^2}{2!}\ddot{x}_j(k) + \cdots \\ \dot{x}_j(k+1) &= \dot{x}_j(k) + \Delta T \ddot{x}_j(k) + \frac{\Delta T^2}{2!}\dddot{x}_j(k) + \cdots \end{aligned} \tag{1}$$

where k is a discrete time index and ΔT is the sampling time.

Similarly, we write the series expansions for position and velocity along the other two orthogonal axes. Now, by neglecting the higher order terms, we obtain an approximate linear state model for each joint j as

$$X_j(k+1) = \mathbf{F}X_j(k) + W(k) \tag{2}$$

where

$$X_j(k) = \begin{bmatrix} x_j(k) \\ \dot{x}_j(k) \\ y_j(k) \\ \dot{y}_j(k) \\ z_j(k) \\ \dot{z}_j(k) \end{bmatrix}, \mathbf{F} = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and}$$

$W(k) = [w_1(k) \quad w_2(k) \quad w_3(k) \quad w_4(k) \quad w_5(k) \quad w_6(k)]^T$ is the system disturbance with a covariance matrix $\mathbf{Q}(k)$. If we assume $w_i(k) = 0$ for all k, then the acceleration and higher order derivatives are zero. This implies that the joint is moving at a constant velocity, which is not reflective of the actual motion of the human. Accordingly, we expect that the filter may not work well. Therefore, we address the question whether we can make it to work sufficiently well by assuming that each $w_i(k)$ is a zero-mean white random process and choosing the values of $\mathbf{Q(k)}$ appropriately. In particular, we model the process covariance terms using the formulation from [29]

$$\mathbf{Q}(k) = E[W(k)W(k)^T] = q\int_{t_k}^{t_{k+1}} F(t_{k+1},\tau)F^T(t_{k+1},\tau)d\tau$$

$$\approx q\Delta T\begin{bmatrix} 1+\Delta T^2 & \Delta T & 0 & 0 & 0 & 0 \\ \Delta T & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+\Delta T^2 & \Delta T & 0 & 0 \\ 0 & 0 & \Delta T & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\Delta T^2 & \Delta T \\ 0 & 0 & 0 & 0 & \Delta T & 1 \end{bmatrix} \tag{3}$$

where q is the strength of the noise.

Note that we obtain only the joint position measurements from each Kinect. Consequently, let $Y_j(k+1) = [x_j^m(k+1)y_j^m(k+1)z_j^m(k+1)]^T$ represent the position measurement[7] for joint j. Now, the measurement model for each joint j is given by

$$Y_j(k+1) = \mathbf{H}X_j(k+1) + V(k+1) \tag{4}$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } V(k+1) = \begin{bmatrix} v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{bmatrix}$$

$V(k+1)$ is the measurement noise with a covariance matrix $\mathbf{R}(k+1)$.

We make the following assumptions with respect to the various noise related variables: (a) Each $w_i(k)$, $(i=1, 2,\ldots,6)$ is a zero-mean white random process. (b) $\mathbf{v}_x(k+1), \mathbf{v}_y(k+1)$, and $\mathbf{v}_z(k+1)$ are independent, zero-mean, and Gaussian noises with variances of $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 0.06\ \text{m}^2$. (c) $W(k)$ and $V(l)$ are uncorrelated for all $k \geq 0$ and $l \geq 0$ [30]. (d) $W(k)$ and $V(k)$ are uncorrelated with the initial state $X(0)$ as their respective sources are different.

Let $X_j^-(k)$ represent the state prediction for kth time step, $X_j'(k)$ represent the corrected state estimate after the measurement is made available, and $K_j(k)$ represent the Kalman gain. Let $\mathbf{P}_j^-(k)$ and $\mathbf{P}_j(k)$ represent the predicted and estimated error covariances in the state, respectively. Now, we implement the distributed discrete Kalman filter to estimate the state for each joint j by using Algorithm 1.

Algorithm 1 Kalman filter implementation for joint j

1: $k=0$;
2: $X_j'(0) = E[X_j(0)] = [x_j^m(0) \quad 0 \quad y_j^m(0) \quad 0 \quad z_j^m(0 \quad 0)]^T$;
3: $\mathbf{P}_j(0) = \mathbf{P}_0$;
4: $k \leftarrow k+1$;
5: $X_j^-(k) = \mathbf{F}X_j'(k-1)$;
6: $\mathbf{P}_j^-(k) = \mathbf{F}\mathbf{P}_j(k-1)\mathbf{F}^T + \mathbf{Q}$;
7: $K_j(k) = \mathbf{P}_j^-(k)\mathbf{H}^T(\mathbf{H}\mathbf{P}_j^-(k)\mathbf{H}^T + \mathbf{R})^{-1}$;
8: $X_j'(k) = X_j^-(k) + K_j(k)(Y_j(k) - \mathbf{H}X_j^-(k))$;
9: $\mathbf{P}_j(k) = (\mathbf{I} - K_j(k)\mathbf{H})\mathbf{P}_j^-(k)$;
10: Go to Step 4;

3.2.2 Data Fusion. As mentioned earlier, the position estimates of each joint j obtained from all the four Kinects are used as inputs to a particle filter [31]. The same state model derived in Eq. (2) is used here. For each joint j, the median of the state estimates $\{X_{1j}'(k), X_{2j}'(k), X_{3j}'(k), X_{4j}'(k)\}$, represented by $X_j^M(k)$, is used as the input to the jth particle filter at time step k. We assume that the measurement noise in each state ψ_i follows a Gaussian distribution with zero mean and variance σ_ψ^2. We can write the measurement model as

$$Y_j(k+1) = X_j(k+1) + V(k+1) \tag{5}$$

Now, we implement a particle filter for joint j using the pseudocode in Algorithm 2.

Algorithm 2 Particle filter implementation for joint j

1: $k=0$;
2: $X_j^M(0) = \text{Median}[X_{1j}'(0), X_{2j}'(0), X_{3j}'(0), X_{4j}'(0)]$;
3: $Y_j(0) = X_j'(0) = X_j(0) = X_j^M(0)$;
4: Initialize N particles $\{\phi_{ij}(0) : i = 1, 2, \ldots, N\}$ from a Gaussian distribution $\mathcal{N}(X_j'(0), \mathbf{Q})$;
5: $k \leftarrow k+1$;
6: $Y_j(k) = \text{Median}[X_{1j}'(k), X_{2j}'(k), X_{3j}'(k), X_{4j}'(k)]$;
7: $\omega_j(k) = 0$;
8: **for** $i = 1 : N$ **do**
9: $\phi_i(k) = F\phi_i(k-1) + GW(k-1)$;
10: $Y_{ij}(k) = \phi_i(k) + V(k)$;
11: $\omega_{ij}(k) = \frac{1}{(2\pi)^3|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(Y_{ij}(k) - Y_j(k))^T\Sigma^{-1}(Y_{ij}(k) - Y_j(k))\right)$;
12: $\omega_j(k) \leftarrow \omega_j(k) + \omega_{ij}(k)$;
13: **end for**
14: Generate a CDF Ω from the set of p.m.f.s assigned to the particles $\{\omega_{ij}(k)/\omega_j(k) : i = 1, 2, \ldots N\}$;
15: Resample the N particles $\{\phi_{ij}(k) : i = 1, 2, \ldots, N\}$ from Ω;
16: $X_j'(k) = (\frac{1}{N})\sum_{i=1}^{N}\phi_{ij}(k)$;
17: Go to Step 5;

3.2.3 Estimation Performance. The tracking performance of the filter is tested by conducting the following experiment: A human moves his wrist from one known point to another known point in the work cell and the measurements from all Kinects are collected and processed using the filtering scheme built around Algorithm 1 and Algorithm 2. The tracking performance along x, y, and z axes is shown in Figs. 4–6, respectively. Each plot includes the ground truth values of initial and final positions, local measurements of the wrist-joint from one of the Kinects, the corresponding local Kalman filter output, the median of all the four Kalman filter outputs, and the particle filter output that provides the final estimate of the wrist-joint motion. Note from Figs. 4 and 6 that the scales used to plot the x and z graphs are different. Therefore, the margins between the measured and estimated values appear to be different in these graphs; but they are indeed similar to each other in reality. A 3D plot of this tracking data is shown in Fig. 7. Note that the particle filter acts upon the median output and provides an improved estimate of the joint motion.

We test the estimation accuracy of the overall sensing system in the following way: A human is made to stand at different randomly selected known positions in the work cell. By assuming different postures at each position, ground truth data for a total of 15 postures are collected for the neck, shoulder, elbow, and wrist joints. Now, we compare this ground truth data to corresponding estimates provided by the sensing system. For illustration purpose, we use six out of these postures that are shown in Fig. 8. The discrepancy between the ground truth and estimated values are shown via projections of the joint positions on the XY plane (Fig. 9) and YZ plane (Fig. 10). In these figures, for each posture, a red-colored * and a green-colored * represent the ground truth and the estimated values for the neck-joint, respectively. Figure 11 shows the discrepancy values for each joint averaged over all the 15 postures. Note that the estimated values match with the ground truth within a margin of $\approx$ 4–5 cm.

4 Precollision Strategy to Achieve Safe HRC

The problem of ensuring safety based on separation monitoring is related to the traditional robot collision avoidance problem.

[7] The Kinect index is omitted for brevity.

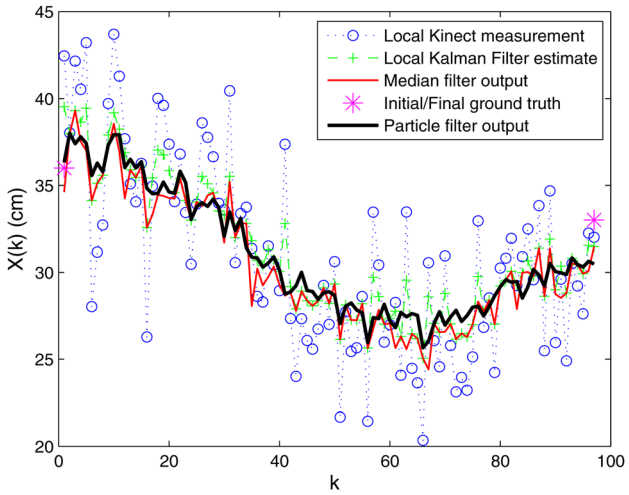

Fig. 4 Filter tracking performance along *x* axis

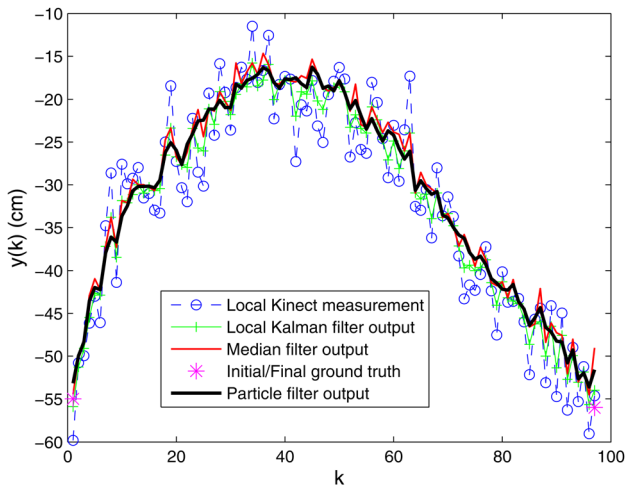

Fig. 5 Filter tracking performance along *y* axis

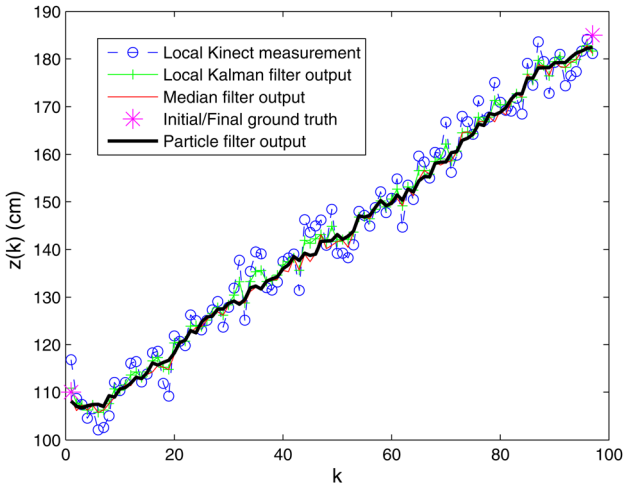

Fig. 6 Filter tracking performance along *z* axis

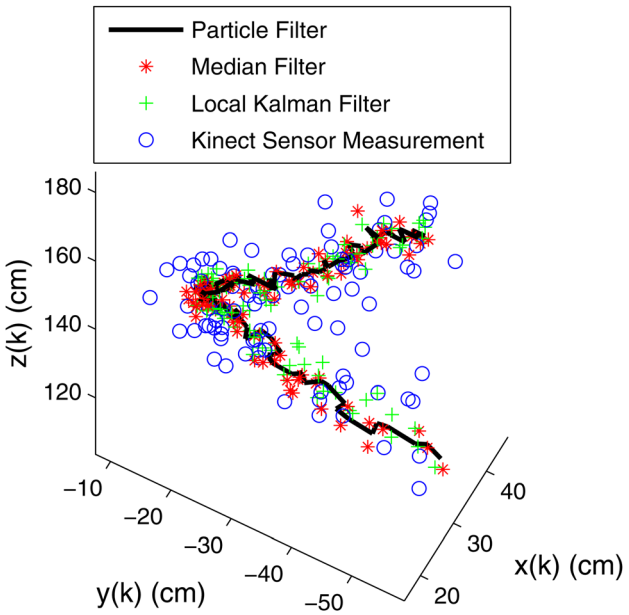

Fig. 7 Filter tracking performance in 3D

However, the properties of physical interaction scenarios in shared work cells significantly differ from classical settings. For example, safety cannot be guaranteed always, if the robot responds to a detected imminent collision by using movements along alternative paths. This is mainly due to the inherently random nature of human motion, which is difficult to predict, and the dynamic nature of the robot implementing such a collision avoidance strategy. In addition, these methods may increase the computational overhead as the system must try to find collision-free paths in real-time. Velocity-scaling based methods [32] address these issues by operating the robot in a trimodal state. In particular, the robot operates in a *clear* (normal functioning) state when the human is far away from it. When the separation between them is less than a specified threshold, the robot transitions into a *slow* state, in which it continues to move in the same path, but at a reduced speed. When the separation is less than a second threshold (whose value is smaller than that of the first one), the robot enters a *pause* state, in which it comes to a safe, controlled stop.

Our approach to ensuring safety while a human and robot collaborate in close proximity with each other consists of pausing the robot's motion whenever an imminent collision between them is detected by the system. This is similar to a simpler bimodal control strategy, in which the robot directly transitions from *clear* to *pause* when the estimated separation is below a threshold distance. This *stop-go* approach to safety is in line with the recommendations put forward by the ISO standard 10218 [33,34].

In order to track the physical separation, the 20-joint human model generated by the exteroceptive sensing system (described in the previous section) is augmented by approximating all pairs of neighboring joints by dynamic bounding spheres that move in a 3D space as a function of the movements performed by the human in real-time. Now, we use a *roll-out* strategy, in which we precompute the robot's trajectory into the near future in order to create a temporal set of robot's postures for the next few seconds and check whether anyone of the postures in this set collides with one of the bounding spheres of the human model. This precollision strategy is implemented in a virtual simulation engine that is developed based on Tundra software.

First, a simulated robot, with a configuration and dimensions that are identical to the physical robot, is instantiated within the virtual environment. The simulated robot replicates the motion of the physical robot in real-time by using the same motor commands that drive the physical robot. The robot's motion plan is assumed to be known beforehand. Therefore, at time $t = 0$, we generate a set of 10 robot's postures by using this information, a sampling time of 0.3 s, and a roll-out parameter of 3 s. This set is

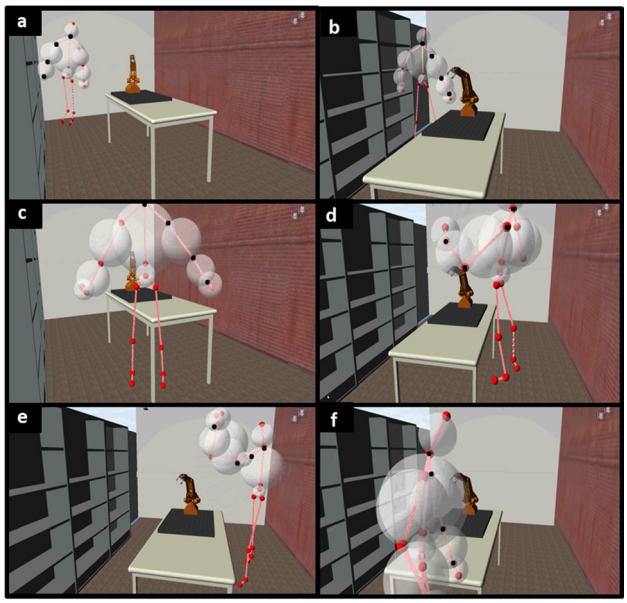

Fig. 8 Postures used to test the estimation accuracy of the overall system

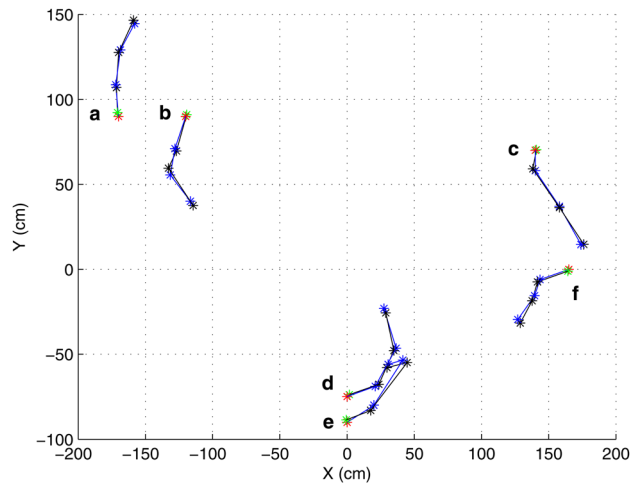

Fig. 9 Discrepancy between projections of ground truth and estimated values on the *XY* plane

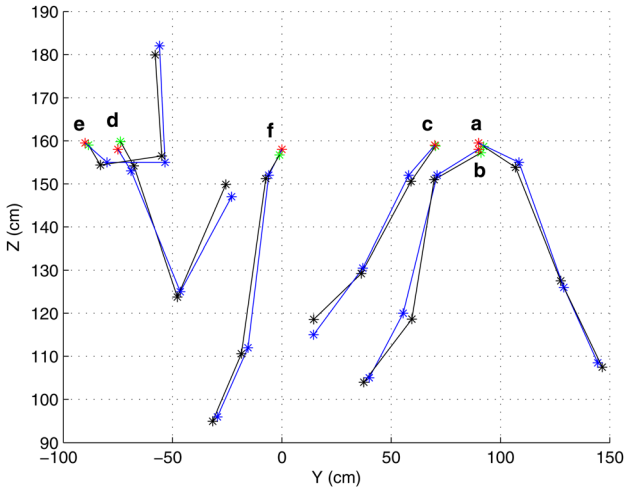

Fig. 10 Discrepancy between projections of ground truth and estimated values on the *YZ* plane

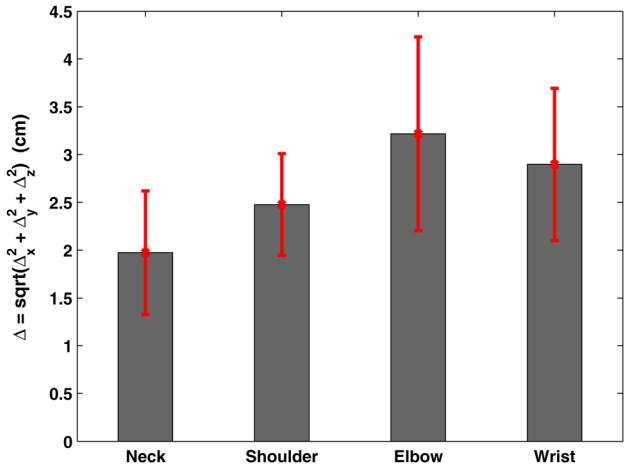

Fig. 11 Discrepancy between ground truth and estimated values for each joint averaged over 15 locations

updated at control-sampling frequency, according to a FIFO method, by removing the robot's current posture from the set and adding its future posture after 3 s to the set.

Second, a simulated human model, with degrees of freedom identical to the one given by the Kinect, is built and instantiated within the same virtual environment. The simulated human model replicates the motion of the refined human model generated by the exteroceptive system by accessing the instantaneous positions of all the 20 joints. Since the joints below the hip do not interfere with the robot during any part of the interaction, they are not considered in the computation of the bounding spheres for the human model.

Figure 12 illustrates the precollision strategy based on the movement of the bounding spheres. From Fig. 12(*a*), the human is in front of the robot when it has just started lifting a part at $t = 0$ s. As there is no intersection between its current set of roll-out postures and the human model, the robot continues its intended task of lifting the part from the table surface. However, at $t = 3$ s (Fig. 12(*b*)), note that the human's hand reaches a state in which a collision is imminent. The roll-out strategy enables the system to detect this condition and pause the robot's motion immediately. It also raises a visual alarm (the sphere changes color from white to red as seen in the figure), which is displayed on a monitor and an audio alarm to alert the human. After $t = 5$ s (Figs. 12(*c*) and 12(*d*)), the robot automatically resumes its task as the human's hand is retrieved into a safety zone.

5 Results

We report results from an experimental scenario, in which a real robot and a human perform a shared assembly task. The physical robot used for the experiments is Lab-Volt 5150 5 DOF manipulator. The task consists of assembling the parts of a simplified chassis assembly consisting of the following parts: Main chassis, a center roll bar, a rear brace, two radio boxes, and four screws. An assembly planning system developed in our earlier work [19] takes a 3D CAD model of the assembly and automatically generates an assembly sequence that drives the task sequence of the robot.

We assign the roles of the human and the robot as follows: Whereas the human picks each part to be assembled and places it in front of the robot, the robot attempts to pick a part available in front of it and proceeds to place it in its intended location in the assembly. The robot motion is kept asynchronous with respect to that of the human on purpose. That is, the robot does not wait to

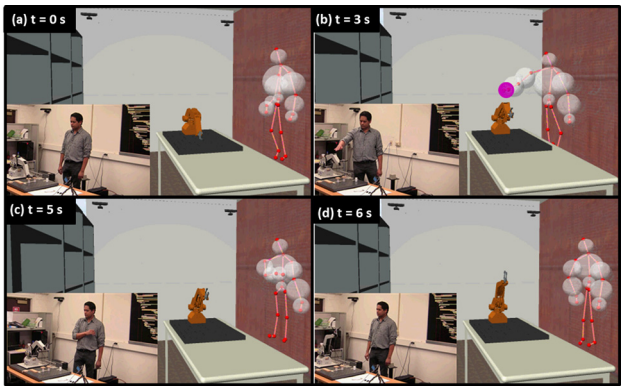

Fig. 12 Illustration of precollision strategy: (*a*) Human is far away from the robot. As the distance between the spheres is significant, robot performs its intended task. (*b*) An imminent collision is detected by the system; therefore, the robot is paused and a visual alarm is raised (bounding spheres change color). (*c*) and (*d*) Human returns to a safety zone; therefore, the robot resumes its motion.

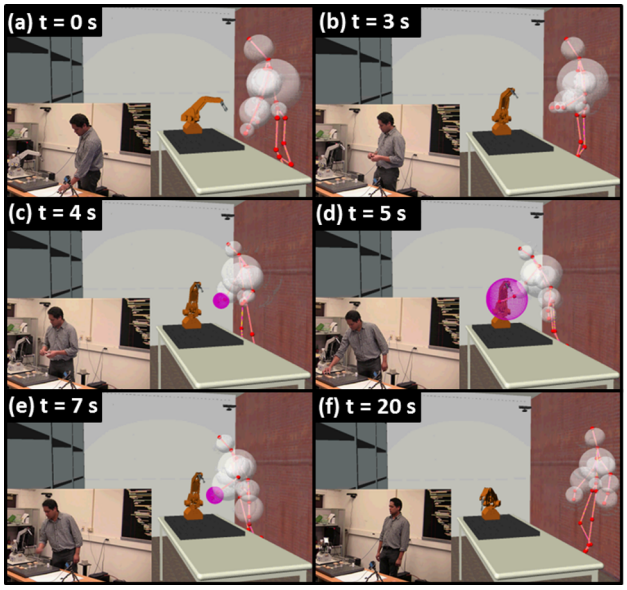

Fig. 13 Robot and human collaborate to assemble the third part (radio box) onto the chassis

reach the part until the human finishes placing it in front of the robot. This thereby sets up an interaction scenario for possible collisions between the human and robot. Figure 13 shows how the robot and human collaborate to assemble one of the parts onto the main chassis. From Fig. 13(*c*), note that the robot pauses its motion when human intervenes to place the part in front of it and resumes its motion when the human turns away Fig. 13(*f*). Similar real-time behavior is observed as the robot and the human collaborate to assemble the remaining parts.

6 Conclusions

We presented a separation monitoring framework that allows a robot and human to safely collaborate and achieve shared tasks in assembly cells. The main contributions of this paper can be summarized as:

(1) Design of an *N*–Kinect framework to generate a 3D model of human's movements in real-time.
(2) Experimental procedure for optimal placement of multiple Kinects in the work cell.
(3) Technique to rapidly evaluate human-robot interference in 3D Euclidean space by using a physics-based simulation engine.
(4) Precollision strategy to achieve safe HRC

In the current work, the precollision strategy consisted of bringing the robot to a complete stop whenever the system detected an intersection between the bounding spheres of the robot and the human. However, the human model based prediction of the human movement can be easily extended to derive better motion goals for the robot, which cater for safety as well as productivity. For example, a trimodal control strategy, in which the robot transitions into an intermediate slow-speed state before coming to a complete stop can be easily implemented by incorporating the velocity estimates of the human model into the robot control algorithm. For this purpose, the current Taylor series based model can be extended to more practical dynamic models without the constant velocity assumption. Using real data obtained from extensive experiments, we demonstrated that our collaborative framework is robust and accurate. However, a more exhaustive evaluation of the accuracy of the tracking system can be made by comparing it with other tracking and motion capture systems. Further, a more rigorous performance analysis of our approach can be carried out based on metrics for separation monitoring that are only starting to appear.

Acknowledgment

This work was supported in part by the National Science Foundation Grant No. CMMI1200087 and by the Center for Energetic Concepts Development. Opinions expressed are those of the authors and do not necessarily reflect opinions of the sponsors.

References

[1] Bicchi, A., and Tonietti, G., 2004, "Fast and Soft Arm Tactics: Dealing With the Safety-Performance Trade-Off in Robot Arms Design and Control," IEEE Rob. Autom. Mag., **11**(2), pp. 22–33.
[2] Haddadin, S., Albu-Schaffer, A., and Hirzinger, G., 2010, "Safety Analysis for a Human-Friendly Manipulator," Int. J. Soc. Robot., **2**, pp. 235–252.
[3] Haddadin, S., Albu-Schaffer, A., and Hirzinger, G., 2009, "Requirements for Safe Robots: Measurements, Analysis and New Insights," Int. J. Robot. Res., **28**(11–12), pp. 1507–1527.
[4] Heinzmann, J., and Zelinsky, A., 2003, "Quantitative Safety Guarantees for Physical Human-Robot Interaction," Int. J. Robot. Res., **22**(7–8), pp. 479–504.
[5] Vogel, C., Poggendorf, M., Walter, C., and Elkmann, N., 2011, "Towards Safe Physical Human-Robot Collaboration: A Projection-Based Safety System," Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 3355–3360.
[6] Balan, L., and Bone, G. M., 2006, "Real-Time 3d Collision Avoidance Method for Safe Human and Robot Coexistence," Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 276–282.
[7] Flacco, F., and De Luca, A., 2010, "Multiple Depth/Presence Sensors: Integration and Optimal Placement for Human/Robot Coexistence," Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 3916–3923.
[8] Flacco, F., Kroger, T., De Luca, A., and Khatib, O., 2012, "A Depth Space Approach to Human-Robot Collision Avoidance," Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 338–345.
[9] Kuhn, S., Gecks, T., and Henrich, D., 2006, "Velocity Control for Safe Robot Guidance Based on Fused Vision and Force/Torque Data," IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 485–492.
[10] Kulic, D., and Croft, E., 2007, "Pre-Collision Safety Strategies for Human-Robot Interaction," Auton. Rob., **22**(2), pp. 149–164.
[11] Najmaei, N., and Kermani, M., 2010, "Prediction-Based Reactive Control Strategy for Human-Robot Interactions," Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 3434–3439.
[12] Schiavi, R., Bicchi, A., and Flacco, F., 2009, "Integration of Active and Passive Compliance Control for Safe Human-Robot Coexistence," Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 259–264.
[13] Haddadin, S., Albu-Schaffer, A., De Luca, A., and Hirzinger, G., 2008, "Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction," Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp. 3356–3363.
[14] Shin, D., Sardellitti, I., Park, Y.-L., Khatib, O., and Cutkosky, M. R., 2010, "Design and Control of a Bio-Inspired Human-Friendly Robot," Int. J. Robot. Res., **29**(5), pp. 571–584.

[15] Marvel, J. A., 2013, "Performance Metrics of Speed and Separation Monitoring in Shared Workspaces," IEEE. Trans. Autom. Sci. Eng., **10**(2), pp. 405–414.
[16] De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A., 2008. "An Atlas of Physical Human–Robot Interaction," Mech. Mach. Theory, **43**(3), pp. 253–270.
[17] Laborie, P., and Ghallab, M., 1995, "Planning With Sharable Resource Constraints," International Joint Conference on Artificial Intelligence, LAWRENCE ERLBAUM ASSOCIATES LTD, Vol. 14, pp. 1643–1651.
[18] Gupta, S. K., Paredis, C., Sinha, R., and Brown, P., 2001, "Intelligent Assembly Modeling and Simulation," Assem. Autom., **21**(3), pp. 215–235.
[19] Morato, C., Kaipa, K. N., and Gupta, S. K., 2013, "Improving Assembly Precedence Constraint Generation by Utilizing Motion Planning and Part Interaction Clusters," Comput.-Aided Des., **45**(11), pp. 1349–1364.
[20] Brough, J., Schwartz, M., Gupta, S. K., Anand, D., Kavetsky, R., and Pettersen, R., 2007, "Towards Development of a Virtual Environment-Based Training System for Mechanical Assembly Operations," Virtual Reality, **11**(4), pp. 189–206.
[21] Gupta, S. K., Anand, D., Brough, J., Kavetsky, R., Schwartz, M., and Thakur, A., 2008, "A Survey of the Virtual Environments-Based Assembly Training Applications," Virtual Manufacturing Workshop, Turin, Italy, October.
[22] Kaipa, K. N., Morato, C., Zhao, B., and Gupta, S. K., 2012, "Instruction Generation for Assembly Operation Performed by Humans," ASME Computers and Information in Engineering Conference, Chicago, IL, August 2–15, 2012.
[23] Smith, C., and Christensen, H. I., 2009, "Wiimote Robot Control Using Human Motion Models," Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, pp. 5509–5515.
[24] Morato, C., Kaipa, K. N., Zhao, B., and Gupta, S. K., 2013, "Safe Human Robot Interaction by using Exteroceptive Sensing Based Human Modeling," ASME Computers and Information in Engineering Conference, Portland, OR, August.
[25] Henrich, D., and Gecks, T., 2008, "Multi-Camera Collision Detection Between Known and Unknown Objects," Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on, pp. 1–10.
[26] Fischer, J., H. B., and Schilling, A., 2007, "Using Time-of-Flight Range Data for Occlusion Handling in Augmented Reality," Eurographics Symposium on Virtual Environments (EGVE).
[27] Valentini, P. P., 2012, "Natural Interface in Augmented Reality Interactive Simulations," Virtual Phys. Prototyping, **7**, pp. 137–151.
[28] Najmaei, N., Kermani, M., and Al-Lawati, M., 2011, "A New Sensory System for Modeling and Tracking Humans Within Industrial Work Cells," IEEE Trans. Instrum. Meas., **60**(4), pp. 1227–1236.
[29] Maybeck, P., 1979, *Stochastic Models, Estimation, and Control*, Academic Press, Inc, New York, Vol. 1.
[30] Corrales, J. A., Candelas, F. A., and Torres, F., 2008, "Hybrid Tracking of Human Operators Using imu/uwb Data Fusion by a Kalman Filter," Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, HRI'08, ACM, pp. 193–200.
[31] Andrieu, C., and Doucet, A., 2002, "Particle Filtering for Partially Observed Gaussian State Space Models," J. R. Stat. Soc., **64**, pp. 827–836.
[32] Davies, S., 2007, "Watching Out for the Workers [Safety Workstations]," Manuf., IET., **86**(4), pp. 32–34.
[33] ISO 10218-1:2011, "Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 1: Robots," International Organization for Standardization, Geneva, Switzerland, 2011. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=51330
[34] ISO 10218-2:2011, "Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 2: Industrial Robot Systems and Integration," International Organization for Standardization, Geneva, Switzerland, 2011. http://www.iso.org/iso/catalogue_detail.htm?csnumber=41571.